

Creating Continents

^

[Contents](#) [Previous](#) [Next](#)

Goal: Guide you through creating external continents.

Before running the tutorial below, type *"python"* or *"cdat"* at the command line. You will see the python prompt appear (i.e., *">>>"*). You can now enter the command lines below.

You can [view](#) or [download](#) the full source code. To run the source code at the command line, type: *"python show_continents.py"*.

EXTERNAL CONTINENTS:

Included with the CDAT/VCS module are three ASCII continents files:

- *'data_continent_states'* ("United States"), continental data for "Fine Continents" and the boundaries of the 48 continental United States;
- *'data_continent_political'* ("Political Borders"), continental data for "Fine Continents" and political borders of sovereign countries;
- *'data_continent_river'* ("Major North America Rivers"), continental data for "Fine Continents" and major North America rivers.

(Note, these continents files may be out of date. Particularly, the "Political Borders" continental file.)

These external continent files are located in the user's: `$HOME/PCMDI_GRAPHICS` directory. User created external continent files must be also placed in this directory.

ADDING YOUR OWN EXTERNAL CONTINENTS:

As many as six additional continental map files can be defined by the user. These files must be located in the user's `$HOME/PCMDI_GRAPHICS` directory and must be named: `data_continent_other7`, `data_continent_other8`, ..., `data_continent_other12`, respectively.

ASCII FORMAT FOR THE EXTERNAL CONTINENTS FILES:

(number of y-x coordinates*2); continental-type value (always 1); values of y-x coordinates

To indicate the end-of-file, a value of `"-99 -99"` is given.

Example of a continents file (e.g., Save the below text in the file `$HOME/PCMDI_GRAPHICS/data_continent_other8`):

```
12 1 64.182 -180.000 69.763 -170.156
64.182 -180.000 64.182 -170.156 66.973 -170.156 66.973 -178.594
69.763 -178.594 69.763 -180.000
282 1 83.705 -18.281 11.162 -164.531

69.763 -164.531 58.602 -164.531 58.602 -158.906 55.811 -158.906
55.811 -156.094 58.602 -156.094 58.602 -147.656 61.392 -147.656
61.392 -144.844 58.602 -144.844 58.602 -136.406 55.811 -136.406
55.811 -130.781 53.021 -130.781 53.021 -127.969 50.230 -127.969
50.230 -125.156 39.068 -125.156 39.068 -122.344 36.278 -122.344
36.278 -119.531 33.487 -119.531 33.487 -116.719 30.697 -116.719
30.697 -113.906 27.906 -113.906 27.906 -111.094 25.115 -111.094
```

```

25.115 -108.281 22.325 -108.281 22.325 -105.469 19.534 -105.469
19.534 -102.656 16.744 -102.656 16.744 -94.219 13.953 -94.219
13.953 -85.781 11.162 -85.781 11.162 -82.969 16.744 -82.969
16.744 -88.594 19.534 -88.594 19.534 -85.781 22.325 -85.781
22.325 -88.594 19.534 -88.594 19.534 -97.031 27.906 -97.031
27.906 -94.219 30.697 -94.219 30.697 -91.406 27.906 -91.406
27.906 -88.594 30.697 -88.594 30.697 -82.969 25.115 -82.969
25.115 -80.156 33.487 -80.156 33.487 -77.344 36.278 -77.344
36.278 -74.531 39.068 -74.531 39.068 -71.719 41.859 -71.719
41.859 -68.906 44.649 -68.906 44.649 -63.281 50.230 -63.281
50.230 -57.656 47.440 -57.656 47.440 -54.844 53.021 -54.844
53.021 -57.656 55.811 -57.656 55.811 -63.281 58.602 -63.281
58.602 -68.906 61.392 -68.906 61.392 -66.094 64.182 -66.094
64.182 -63.281 66.973 -63.281 66.973 -66.094 69.763 -66.094
69.763 -71.719 72.553 -71.719 72.553 -77.344 78.131 -77.344
78.131 -68.906 80.919 -68.906 80.919 -66.094 78.131 -66.094
78.131 -68.906 75.342 -68.906 75.342 -57.656 72.553 -57.656
72.553 -54.844 66.973 -54.844 66.973 -52.031 64.182 -52.031
64.182 -49.219 61.392 -49.219 61.392 -40.781 64.182 -40.781
64.182 -37.969 66.973 -37.969 66.973 -26.719 69.763 -26.719
69.763 -21.094 72.553 -21.094 72.553 -18.281 80.919 -18.281
80.919 -23.906 83.705 -23.906 83.705 -52.031 80.919 -52.031
80.919 -63.281 83.705 -63.281 83.705 -85.781 80.919 -85.781
80.919 -97.031 78.131 -97.031 78.131 -99.844 69.763 -99.844
69.763 -102.656 72.553 -102.656 72.553 -116.719 69.763 -116.719
69.763 -119.531 72.553 -119.531 72.553 -116.719 75.342 -116.719
75.342 -113.906 78.131 -113.906 78.131 -116.719 75.342 -116.719
75.342 -125.156 72.553 -125.156 72.553 -122.344 69.763 -122.344
69.763 -164.531
-99 -99

```

The first line specifies the connection of six points of (latitude, longitude) coordinates in the upper left corner of the plot.

The second line of this example specifies the connection of 141 points, which represents a very rough outline of North America.

The entry "`-99 -99`" indicate that no more data are to be read in (i.e., denotes end-of-file).

Note the total number of points for any given line should not exceed 200 points.

```

# Import the modules needed for the tutorial
# Import the modules needed for the tutorial
# cdms - Climate Data Management system accesses gridded data.
# vcs - Visualization and control System 1D and 2D plotting routines.
# cdutil - Climate utilizes that contains miscellaneous routines for
#           manipulating variables.
# time - This module provides various functions to mainpulate time values.
# os - Operation System routines for Mac, DOS, NT, or Posix depending on
#       the system you're on.
# sys - This module provides access to some objects used or maintained by
#       the interpreter and to functions that interact strongly with the interpreter.
import vcs, cdms, cdutil, time, os, sys

```

```

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set and get a subset of the time dimension
data = cdmsfile('clt', longitude=(-180, 180), latitude = (-90., 90.))

# Initial VCS:
v = vcs.init()

#Plot the data using the created data_continent_other8 file.
v.plot( data, continents=7 )

```

USING THE NEWLY CREATED CONTINENTS:

The continent-type values in the CDAT API range from 0 to 11, where:

- 0 signifies "No Continents"
- 1 signifies "Fine Continents"
- 2 signifies "Coarse Continents"
- 3 signifies "United States Continents"
- 4 signifies "Political Borders Continents"
- 5 signifies "North American Rivers Continents"

Values 6 through 11 signify the continents files data_continent_other7 through data_continent_12, respectively.

EXAMPLE PYTHON SCRIPT SHOWING CONTINENTS:

```

# Import the modules needed for the tutorial
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set and get a subset of the time dimension
data = cdmsfile('clt', longitude=(-180, 180), latitude = (-90., 90.))

continent_type = { 0: "0 signifies - No Continents",
 1: "1 signifies - Fine Continents",
 2: "2 signifies - Coarse # Continents",
 3: "3 signifies - United States Continents",
 4: "4 signifies - Political Borders Continents",
 5: "5 signifies - North American Rivers Continents",
 6: "6 signifies - User continent file data_continent_other7",
 7: "7 signifies - User continent file data_continent_other8",
 8: "8 signifies - User continent file data_continent_other9",
 9: "9 signifies - User continent file data_continent_other10",
10: "10 signifies - User continent file data_continent_other11",
11: "11 signifies - User continent file data_continent_other12"
}

# Initial VCS:
v = vcs.init()

for i in range( 12 ):
print "continent_type ", continent_type[i]
v.plot(data, continents=i)
time.sleep(2)

```

v.clear()

EXAMPLE from VCDAT:

From VCDAT, under the graphics "Options" menu button, you will see the "Continents Types" pull right menu item. All of the continents (including the user defined continents) are selectable from here.

MORE INFORMATION:

There is more information on how to set the continents line type, line color, etc. in other sections of the tutorial. In VCDAT, you can modify the continents from the "Set Graphics Method Attributes..." menu panel.

ANOTHER ASCII FILE EXAMPLE:

In your \$HOME/PCMDI_GRAPHICS directory, create the file "data_continents_other9" and enter the following lines to. Then run the Python and VCDAT examples above. (This ASCII file will draw a coarse continental outline of North America and Australia.

```
Â 282Â Â 1Â 83.705 -18.281 11.162 -164.531
69.763 -164.531Â Â 58.602 -164.531Â 58.602Â -158.906 55.811 -158.906
55.811 -156.094Â Â 58.602 -156.094Â 58.602Â -147.656 61.392 -147.656
61.392 -144.844Â Â 58.602 -144.844Â 58.602Â -136.406 55.811 -136.406
55.811 -130.781Â Â 53.021 -130.781Â 53.021Â -127.969 50.230 -127.969
50.230 -125.156Â Â 39.068 -125.156Â 39.068Â -122.344 36.278 -122.344
36.278 -119.531Â Â 33.487 -119.531Â 33.487Â -116.719 30.697 -116.719
30.697 -113.906Â Â 27.906 -113.906Â 27.906Â -111.094 25.115 -111.094
25.115 -108.281Â Â 22.325 -108.281Â 22.325Â -105.469 19.534 -105.469
19.534 -102.656Â Â 16.744 -102.656Â 16.744Â Â -94.219 13.953Â -94.219
13.953Â -85.781Â Â 11.162Â -85.781Â 11.162Â Â -82.969 16.744Â -82.969
16.744Â -88.594Â Â 19.534Â -88.594Â 19.534Â Â -85.781 22.325Â -85.781
22.325Â -88.594Â Â 19.534Â -88.594Â 19.534Â Â -97.031 27.906Â -97.031
27.906Â -94.219Â Â 30.697Â -94.219Â 30.697Â Â -91.406 27.906Â -91.406
27.906Â -88.594Â Â 30.697Â -88.594Â 30.697Â Â -82.969 25.115Â -82.969
25.115Â -80.156Â Â 33.487Â -80.156Â 33.487Â Â -77.344 36.278Â -77.344
36.278Â -74.531Â Â 39.068Â -74.531Â 39.068Â Â -71.719 41.859Â -71.719
41.859Â -68.906Â Â 44.649Â -68.906Â 44.649Â Â -63.281 50.230Â -63.281
50.230Â -57.656Â Â 47.440Â -57.656Â 47.440Â Â -54.844 53.021Â -54.844
53.021Â -57.656Â Â 55.811Â -57.656Â 55.811Â Â -63.281 58.602Â -63.281
58.602Â -68.906Â Â 61.392Â -68.906Â 61.392Â Â -66.094 64.182Â -66.094
64.182Â -63.281Â Â 66.973Â -63.281Â 66.973Â Â -66.094 69.763Â -66.094
69.763Â -71.719Â Â 72.553Â -71.719Â 72.553Â Â -77.344 78.131Â -77.344
78.131Â -68.906Â Â 80.919Â -68.906Â 80.919Â Â -66.094 78.131Â -66.094
78.131Â -68.906Â Â 75.342Â -68.906Â 75.342Â Â -57.656 72.553Â -57.656
72.553Â -54.844Â Â 66.973Â -54.844Â 66.973Â Â -52.031 64.182Â -52.031
64.182Â -49.219Â Â 61.392Â -49.219Â 61.392Â Â -40.781 64.182Â -40.781
64.182Â -37.969Â Â 66.973Â -37.969Â 66.973Â Â -26.719 69.763Â -26.719
69.763Â -21.094Â Â 72.553Â -21.094Â 72.553Â Â -18.281 80.919Â -18.281
80.919Â -23.906Â Â 83.705Â -23.906Â 83.705Â Â -52.031 80.919Â -52.031
80.919Â -63.281Â Â 83.705Â -63.281Â 83.705Â Â -85.781 80.919Â -85.781
80.919Â -97.031Â Â 78.131Â -97.031Â 78.131Â Â -99.844 69.763Â -99.844
69.763 -102.656Â Â 72.553 -102.656Â 72.553Â -116.719 69.763 -116.719
69.763 -119.531Â Â 72.553 -119.531Â 72.553Â -116.719 75.342 -116.719
75.342 -113.906Â Â 78.131 -113.906Â 78.131Â -116.719 75.342 -116.719
75.342 -125.156Â Â 72.553 -125.156Â 72.553Â -122.344 69.763 -122.344
69.763 -164.531
Â Â Â 66Â Â Â Â Â Â Â 1Â Â -36.278Â 113.906 -11.162Â 153.281
-36.278Â 136.406 -36.278Â 142.031 -39.068Â 142.031 -39.068Â 150.469
-33.487Â 150.469 -33.487Â 153.281 -25.115Â 153.281 -25.115Â 150.469
-19.534Â 150.469 -19.534Â 147.656 -16.744Â 147.656 -16.744Â 144.844
-13.953Â 144.844 -13.953Â 142.031 -16.744Â 142.031 -16.744Â 136.406
-11.162Â 136.406 -11.162Â 130.781 -13.953Â 130.781 -13.953Â 125.156
```

-16.744 125.156 -16.744 122.344 -19.534 122.344 -19.534 116.719
-22.325 116.719 -22.325 113.906 -30.697 113.906 -30.697 116.719
-36.278 116.719 -36.278 119.531 -33.487 119.531 -33.487 136.406
-36.278 136.406
-99 -99

CONVERTING Global Mapping Tool (GMT) COASTLINES INTO VCS FORMAT (written by Julien Vienne):

A link to the full source code (i.e., gmt2vcs_continents.py) can be found [*here*](#).

^

[Contents](#) [Previous](#) [Next](#)